

10-27-00 EK 538216 773US

A

JC840 U.S. PTO  
10/26/00

**CERTIFICATE OF MAILING BY "EXPRESS MAIL" UNDER 37 CFR § 1.10**

"Express Mail" mailing label number \_\_\_\_\_

Date of Mailing: \_\_\_\_\_

I hereby certify that the documents indicated below are being deposited with the United States Postal Service under 37 CFR 1.10 on the date indicated above and are addressed to Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231, and mailed on the above Date of Mailing with the above "Express Mail" mailing label number

(Typed or printed name of person mailing paper or fee) \_\_\_\_\_ SIGNATURE of person mailing paper or fee \_\_\_\_\_

JC931 U.S. PTO  
09/697446  
10/26/00

BOX PATENT APPLICATION  
ASSISTANT COMMISSIONER FOR PATENTS  
WASHINGTON, D. C. 20231

DOCKET NUMBER: AUS9-2000-0501-US1  
10/26/00

Sir:

Transmitted herewith for filing is the Patent Application of:

Inventors: David B. Kumhyr et al.

For: IDENTIFYING NON-EXTERNALIZED TEXT STRINGS THAT ARE NOT HARD-CODED

Enclosed are:

- ☒ Patent Specification and Declaration
- ☒ 2 sheets of drawing(s) *informal*
- ☒ An assignment of the invention to International Business Machines Corporation (includes Recordation Form Cover Sheet).
- ☐ A certified copy of a \_\_\_ application
- ☐ An associate power of attorney
- ☐ Information Disclosure Statement, PTO 1449 and copies of references.

The filing fee has been calculated as shown below.

For	Number Filed	Number Extra	Rate	Fee
Basic Fee				\$ 710.00
Total Claims	24 - 20	4	x 18 =	\$ 72.00
Indep. Claims	3 - 3	0	x 80 =	\$ - 0 -
<input type="checkbox"/> MULTIPLE DEPENDENT CLAIM(S) PRESENTED			+ 270 =	\$ - 0 -
			<b>TOTAL</b>	<b>\$ 782.00</b>

- ☒ Please charge my Deposit Account No. 09-0447 in the amount of \$782.00. **A duplicate copy of this sheet is enclosed.**
- ☒ The Assistant Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. 09-0447. **A duplicate copy of this sheet is enclosed.**
  - ☒ Any additional filing fees required under 37 CFR §1.16
  - ☒ Any patent application processing fees under 37 CFR §1.17.

Respectfully submitted,

By:

*Jeffrey S. LaBaw*

Jeffrey S. LaBaw  
Registration No. 31,633  
IBM Corporation  
Intellectual Property Law Dept.  
Internal Zip 9444  
11400 Burnet Road  
Austin, Texas 78758  
Telephone: (512) 823-0496

\\ODMA\PCDOCS\AUSTIN\_1\149945\1  
207:7047-P375US

JEL/WSM

AUS9-2000-0501 US1

PATENT

**IDENTIFYING NON-EXTERNALIZED TEXT STRINGS  
THAT ARE NOT HARD-CODED**

5 **CROSS REFERENCE TO RELATED APPLICATIONS**

The present invention is related to the following U.S. Patent Applications which are incorporated herein by reference:

10 Serial No. \_\_\_\_\_ (Attorney Docket No. AUS9-2000-0499-US1) entitled "Detection of Resource Exceptions" filed \_\_\_\_\_.

Serial No. \_\_\_\_\_ (Attorney Docket No. AUS9-2000-0498-US1) entitled "Pre-Processing Code to Identify and Modify Format of Keys" filed \_\_\_\_\_.

15 **TECHNICAL FIELD**

The present invention relates to the field of internationalization, and more particularly to a scanning program that identifies non hard-coded text strings.

20 **BACKGROUND INFORMATION**

Internationalization is a process of enabling a program, e.g., Java, to run internationally. That is, an internationalized program has the flexibility to run correctly in any country. An internationalized program must be able to read, write and manipulate

localized text. Furthermore, an internationalized program must conform to local customs when displaying dates and times, formatting numbers and sorting strings.

5 Internationalization is becoming increasingly important with the explosive growth of the Internet and the World Wide Web where an ever increasing number of computer users are from various locales. A locale represents a geographic, cultural or political region. One of the problems with internationalization involves the use of text strings that may be hard-coded in the program, e.g., Java. Hard-coded text strings refer to text that will not vary with the locale. That is, the text strings may appear in English even when the program is run on the French locale. Various object-oriented languages such as Java have developed tools to assist in developing internationalized programs and allowing text strings to appear in the language of the locale. A discussion of object-oriented programming languages and in particular Java is deemed appropriate.

10 In an object-oriented programming language such as Java, a class is a collection of data and methods that operate on that data. The data and methods taken together describe the state and behavior of what is commonly referred to as an object. An object in essence includes data and code where the code manipulates the data. Hence a software application may be written using an object-oriented programming language such as Java whereby the program's functionality is implemented using objects.

15 Unlike many programming languages, Java is compiled into machine independent code commonly referred to as bytecodes instead of machine dependent code, i.e. executable code. Bytecodes are stored in a particular file format commonly referred to as a "class file" that includes bytecodes for methods of a class. In addition to the bytecodes for methods of a class, the class file includes a symbol file as well as other ancillary information.

5 A computer program embodied as Java bytecodes in one or more class files is platform independent. The computer program may be executed, unmodified, on any computer that is able to run an implementation of what is commonly referred to as a Java virtual machine. The Java virtual machine is not an actual hardware platform, but rather  
10 a low level software emulator that can be implemented on many different computer processor architectures and under many different operating systems. The Java virtual machine reads and interprets each bytecode so that the instructions may be executed by the native processor. Hence a Java bytecode is capable of functioning on any platform that has a Java virtual machine implementation available. However, bytecode interpretation detracts from processor performance since the microprocessor has to spend some of its processing time interpreting bytecode instructions. Compilers commonly referred to as "just in time (JIT)" were developed to improve the performance of Java virtual machines. A JIT compiler translates Java bytecodes into the processor's native machine code during runtime. The processor then executes the compiled native machine  
15 code.

20 As stated above Java has developed tools to assist in developing internationalized programs and allowing text strings to appear in the language of the locale. One such tool is the use of resource files commonly referred to in Java as resource bundles. A resource bundle class may be used for externalizing text strings, i.e. not hard-coding strings in the program. The resource bundle class represents a bundle of resources that may be looked up by name. The resources may include appropriate text strings for a given locale that are indexed by what are commonly referred to as keys. Keys are free formatted strings that appear in the program code as well as in the resource bundle thereby allowing the program to access the externalized string. By having resource bundles associated with

particular locales, e.g., a resource file with resources associated with the US English locale, a resource file with resources associated with the French locale and so forth, appropriate text strings associated with the particular locale may be loaded at runtime.

5 However, software developers may still hard-code their strings into their application instead of externalizing them and loading them from the resource bundle. Various scanning programs have been developed which attempt to detect hard-coded strings. Unfortunately, these scanning programs simply detect as hard-coded strings all  
10 text enclosed within double quotes (") which are used as string delimiters in Java (as well as other programming languages). However, not all text enclosed within double quotes are hard-coded strings. The text enclosed within the double quotes may be a path name to a resource file, e.g., resource bundle.

It would therefore be desirable to develop a scanning program that identifies non-externalized strings, e.g., path names to resource files, that are not hard-coded but that are enclosed within string delimiters.

**SUMMARY**

5 The problems outlined above may at least in part be solved in some embodiments by a scanning program that scans a code, e.g., Java, line by line until a pair of string delimiters is identified. Once a pair of string delimiters is identified, the scanning program determines whether the string within the pair of string delimiters identified is a path name to a resource file, e.g., resource bundle. If the string is a path name to the resource file, then the string is not a hard-coded string. If the string is not a path name to the resource file, then the string may be identified as a possible hard-coded string.

10 In one embodiment, a method for identifying non-externalized strings that are not hard-coded comprises the step of scanning a code for a pair of string delimiters. The method further comprises the step of determining whether the string within the pair of string delimiters identified is a path name to a resource file. If the string is a path name to the resource file, then the string is a non-externalized string that is not hard-coded. If  
15 the string is not a path name to the resource file, then the string may be identified as a possible hard-coded string.

20 The foregoing has outlined rather broadly the features and technical advantages of the present invention in order that the detailed description of the invention that follows may be better understood. Additional features and advantages of the invention will be described hereinafter which form the subject of the claims of the invention.

**BRIEF DESCRIPTION OF THE DRAWINGS**

5 A better understanding of the present invention can be obtained when the following detailed description is considered in conjunction with the following drawings, in which:

Figure 1 illustrates a data processing system configured in accordance with the present invention; and

Figure 2 is a flowchart of a method for identifying non-externalized strings that are not hard-coded.

**DETAILED DESCRIPTION**

5 The present invention comprises a method, computer program product and data processing system for identifying non-externalized strings that are not hard-coded. In one embodiment of the present invention a scanning program scans a code, e.g., Java, line by line until a pair of string delimiters is identified. Once a pair of string delimiters is identified, the scanning program determines whether the string within the pair of string delimiters identified is a path name to a resource file, e.g., resource bundle. If the string is a path name to the resource file, then the string is a non-externalized string that is not hard-coded. If the string is not a path name to the resource file, then the string may be identified as a possible hard-coded string. It is noted that the even though the following discusses the present invention in conjunction with a Java programming environment the present invention may be implemented in any type of programming environment where the programming language has the capability of externalizing text strings in resource files.

Figure 1 - Computer System

20 Figure 1 illustrates a typical hardware configuration of data processing system 13 which is representative of a hardware environment for practicing the present invention. Data processing system 13 has a central processing unit (CPU) 10, such as a conventional microprocessor, coupled to various other components by system bus 12. An operating system 40, e.g., DOS, OS/2™, runs on CPU 10 and provides control and coordinates the function of the various components of Figure 1. An object-oriented programming



5 system, such as Java 42, runs in conjunction with operating system 40 and provides  
output calls to operating system 40 which implements the various functions to be  
performed by the application 42. Read only memory (ROM) 16 is coupled to system bus  
12 and includes a basic input/output system ("BIOS") that controls certain basic functions  
10 of data processing system 13. Random access memory (RAM) 14, I/O adapter 18, and  
communications adapter 34 are also coupled to system bus 12. It should be noted that  
software components including operating system 40 and application 42 are loaded into  
RAM 14 which is the computer system's main memory. I/O adapter 18 may be a small  
computer system interface ("SCSI") adapter that communicates with disk units 20, e.g.,  
15 disk drive, and tape drives 40. It is noted that the scanning program of the present  
invention that identifies non-externalized strings that are not hard-coded may reside in  
disk unit 20 or in application 42. Communications adapter 34 interconnects bus 12 with  
an outside network enabling data processing system 13 to communication with other  
such systems. Input/Output devices are also connected to system bus 12 via a user  
20 interface adapter 22 and a display adapter 36. Keyboard 24, trackball 28, mouse 26 and  
speaker 30 are all interconnected to bus 12 through user interface adapter 22. Event data  
may be input to the object-oriented programming system through any of these devices.  
A display monitor 38 is connected to system bus 12 by display adapter 36. In this  
manner, a user is capable of inputting to system 13 through keyboard 24, trackball 28  
or mouse 26 and receiving output from system 13 via display 38 or speaker 30.

Preferred implementations of the invention include implementations as a  
computer system programmed to execute the method or methods described herein, and  
as a computer program product. According to the computer system implementations,  
sets of instructions for executing the method or methods are resident in the random

access memory 14 of one or more computer systems configured generally as described above. Until required by the computer system, the set of instructions may be stored as a computer program product in another computer memory, for example, in disk drive 20 (which may include a removable memory such as an optical disk or floppy disk for eventual use in disk drive 20). Furthermore, the computer program product can also be stored at another computer and transmitted when desired to the user's work station by a network or by an external network such as the Internet. One skilled in the art would appreciate that the physical storage of the sets of instructions physically changes the medium upon which it is stored so that the medium carries computer readable information. The change may be electrical, magnetic, chemical or some other physical change.

Figure 2 - Method For Identifying Non-Externalized Strings That Are Not Hard-Coded

Figure 2 illustrates a method 200 for identifying non-externalized strings that are not hard-coded. As stated in the Background Information section, software developers may hard-code their strings into their application, e.g., Java, instead of externalizing them and loading them from the external resource file, e.g., resource bundle. Various scanning programs have been developed which attempt to detect hard-coded strings. Unfortunately, these scanning programs simply detect as hard-coded strings all text enclosed within double quotes, i.e. string delimiters. However, not all text enclosed within double quotes are hard-coded strings. The text enclosed within the double quotes may be a path name to a resource file, e.g., resource bundle. Method 200 is a method

that identifies non-externalized strings, e.g., path names to resource files, that are not hard-coded that are enclosed within string delimiters.

In step 210, a scanning program scans the code of an application program 42 line by line for string delimiters until the scanning program identifies a pair of string delimiters. String delimiters refers to the quotes (") that mark the beginning and end of a text string. For example, in the classic Hello World program written in Java as shown below

```
public class HelloWorld {  
    public static void main  
    (String args[]) {  
        System.out.println("Hello World");  
    }  
}
```

the string delimiters mark the beginning and end of the text string "Hello World."

A determination is then made in step 220 as to whether the scanning program has identified a pair of string delimiters that mark the beginning and end of a text string. If the scanning program has not identified a pair of string delimiters, then method 200 is terminated at step 230. If the scanning program has identified a pair of string delimiters, the scanning program determines whether the string within the string delimiters is a path name to the resource file in step 240. As stated in the Background Information section, a resource file is commonly referred to as a resource bundle in Java. It is further noted that a path name to a resource file, e.g., resource bundle, is a non-externalized string that

is not hard-coded but is enclosed within double quotes within a program, e.g., Java. For example, in the Java code shown below

5       rbCal=ResourceLoader.getBundle("com.tivoli.uif.Resources.CalendarResources");  
      setTitle(ResourceLoader.getString(rbCal, "Holiday Title");

10       the string "com.tivoli.uif.Resources.CalendarResources" is a path name to a resource bundle. Path names to resource bundles are commonly referred to as uniform resource locator (URL). URL's are commonly identified by their dotted signature. It is noted that rbCal is a resource bundle where the method ResourceLoader retrieves calendar resources from the URL "com.tivoli.uif.Resources.CalendarResources." It is further noted that the second line of the above written Java code sets the title to the window to an externalized string "Holiday Title" located in the resource bundle, rbCal. That is, the method ResourceLoader will retrieve the proper string associated with the language of the locale. For example, if the locale is an American locale, then the holiday title on the window may appear as "Christmas." If the locale is a Spanish locale, then the holiday title on the window may appear as "Navidad."

15       Referring to the above example of Java code, the scanning program may identify the string delimiters that mark the beginning and end of the text string "com.tivoli.uif.Resources.CalendarResources" in step 220. In step 240, a determination is made by the scanning program as to whether the string within the string delimiters identified in step 220 is a non-externalized string that is not hard-coded. If the scanning program determines that the string within the string delimiter is a non-externalized string that is not hard-coded, then the scanning program will not flag the string as a possible  
20

5 hard-coded string in step 250. As stated above, the scanning program will identify the string within the string delimiters as a non-externalized string that is not hard-coded if the string is a URL identified from its dotted signature, e.g., "com.tivoli.uif.Resources.CalendarResources." A determination is then made in step 260  
10 as to whether there is any more code to scan by the scanning program. If there is no more code to scan by the scanning program, then method 200 is terminated at step 230. If there is more code to scan by the scanning program, then method 200 proceeds to scan the remaining code for string delimiters in step 210. It is noted for clarity that if the scanning program has identified a pair of string delimiters in a particular line of code in step 220 and there is more code within that particular line, then the scanning program may continue to scan the remainder of that particular line of code and the remaining line(s) of code until the scanning program identifies a pair of string delimiters.

15 If the scanning program determines in step 240 that the string within the string delimiter is not a path name to a resource file, e.g., resource bundle, then the scanning program identifies, i.e. flags, the string as a possible hard-coded string in step 270. For example, in the second line of code of the above example, the string "Holiday Title" may not be identified as a path name in step 240 because "Holiday Title" does not exhibit a dotted signature. Therefore, the string "Holiday Title" may be identified as a possible hard-coded string in step 270. A determination is then made in step 260 as to whether  
20 there is any more code to scan by the scanning program. If there is no more code to scan by the scanning program, then method 200 is terminated at step 230. If there is more code to scan by the scanning program, then method 200 proceeds to scan the remaining code for string delimiters in step 210. It is noted for clarity that if the scanning program has identified a pair of string delimiters in a particular line of code in step 220 and there

is more code within that particular line, then the scanning program may continue to scan the remainder of that particular line of code and the remaining line(s) of code until the scanning program identifies a pair of string delimiters.

It is noted that the scanning program may reside in disk unit 20 or application 42.

5 It is further noted that the scanning program of the present invention may be implemented to detect non-externalized strings that are not hard-coded in any type of programming language that has the capability of externalizing text strings in resource files.

10 Although the method, computer program product and data processing system of the present invention is described in connection with several embodiments, it is not intended to be limited to the specific forms set forth herein, but on the contrary, it is intended to cover such alternatives, modifications, and equivalents, as can be reasonably included within the spirit and scope of the invention as defined by the appended claims. It is noted that the headings are used only for organizational purposes and not meant to  
15 limit the scope of the description or claims.

**CLAIMS:**

1. A method for identifying non-externalized strings that are not hard-coded comprising the steps of:
  - scanning a code for a first pair of string delimiters; and
  - determining whether a string within said first pair of string delimiters is a path name to a resource file;
  - wherein if said string is a path name to said resource file then said string is a non-externalized string that is not hard-coded.
2. The method as recited in claim 1 further comprising the step of:
  - identifying said string as a possible hard-coded string if said string is not a path name to said resource file.
3. The method as recited in claim 1, wherein said code comprises Java code.
4. The method as recited in claim 1, wherein said path name is a uniform resource locator.
5. The method as recited in claim 1, wherein said resource file is a resource bundle.

1        6.        The method as recited in claim 1, wherein said string within said first pair of  
2 string delimiters is a path name to said resource file if said string is in a dot delimited  
3 notation.

1        7.        The method as recited in claim 1, wherein said code is scanned line by line until  
2 said first pair of string delimiters is identified.

1        8.        The method as recited in claim 7, wherein if there is any more code to be scanned  
2 after said first pair of string delimiters is identified, then the method further comprises  
3 the step of:

4                continuing to scan said code for a second pair of string delimiters.



1 9. A computer program product in a computer readable medium for identifying  
2 non-externalized strings that are not hard-coded, comprising:

3 programming operable for scanning a code for a first pair of string delimiters; and  
4 programming operable for determining whether a string within said first pair of  
5 string delimiters is a path name to a resource file;

6 wherein if said string is a path name to said resource file then said string is a  
7 non-externalized string that is not hard-coded.

10. The computer program product as recited in claim 9 further comprises:  
programming operable for identifying said string as a possible hard-coded string  
if said string is not a path name to said resource file.

11. The computer program product as recited in claim 9, wherein said code comprises  
Java code.

12. The computer program product as recited in claim 9, wherein said path name is  
a uniform resource locator.

13. The computer program product as recited in claim 9, wherein said resource file  
is a resource bundle.

14. The computer program product as recited in claim 9, wherein said string within  
said first pair of string delimiters is a path name to said resource file if said string is in  
a dot delimited notation.

1        15.    The computer program product as recited in claim 9, wherein said code is  
2 scanned line by line until said first pair of string delimiters is identified.

1        16.    The computer program product as recited in claim 15, wherein if there is any  
2 more code to be scanned after said first of string delimiters is identified, then the  
3 computer program product further comprises:

4                programming operable for continuing to scan said code for a second pair of string  
5 delimiters.

1 17. A data processing system, comprising:  
2 a processor; and  
3 a memory unit for storing instructions of said processor;  
4 an input mechanism;  
5 an output mechanism;  
6 a bus system for coupling the processor to the memory unit, input mechanism,  
7 and output mechanism;  
8 means for scanning a code for a first pair string delimiters; and  
9 means for determining whether a string within said first pair of string delimiters  
10 is a path name to a resource file;  
11 wherein if said string is a path name to said resource file then said string is a  
12 non-externalized string that is not hard-coded.

1 18. The data processing system as recited in claim 17, wherein the system further  
2 comprises:  
3 means for identifying said string as a possible hard-coded sting if said string is  
4 not a path name to said resource file.

1 19. The data processing system as recited in claim 17, wherein said code comprises  
2 Java code.

1 20. The data processing system as recited in claim 17, wherein said path name is a  
2 uniform resource locator.

1 21. The data processing system as recited in claim 17, wherein said resource file is  
2 a resource bundle.

1 22. The data processing system as recited in claim 17, wherein said string within said  
2 first pair of string delimiters is a path name to said resource file if said string is in a dot  
3 delimited notation.

1 23. The data processing system as recited in claim 17, wherein said code is scanned  
2 line by line until said first pair of string delimiters is identified.

1 24. The data processing system as recited in claim 23, wherein if there is any more  
2 code to be scanned after said first pair of string delimiters is identified, then the system  
3 further comprises:

4 means for continuing to scan said code for a second pair of string delimiters.

## **IDENTIFYING NON-EXTERNALIZED TEXT STRINGS THAT ARE NOT HARD-CODED**

### **ABSTRACT**

5

A method, computer program product and data processing system for identifying non-externalized strings that are not hard-coded. In one embodiment, a method comprises the step of scanning a code, e.g., Java, line by line until a pair of string delimiters is identified. Once a pair of string delimiters is identified, the method further comprises the step of determining whether the string within the pair of string delimiters identified is a path name to a resource file, e.g., resource bundle. If the string is a path name to the resource file, then the string is a non-externalized string that is not hard-coded. If the string is not a path name to the resource file, then the string may be identified as a possible hard-coded string.

10  
15  
20  
25  
30  
35  
40  
45  
50  
55  
60  
65  
70  
75  
80  
85  
90  
95  
100  
105  
110  
115  
120  
125  
130  
135  
140  
145  
150  
155  
160  
165  
170  
175  
180  
185  
190  
195  
200  
205  
210  
215  
220  
225  
230  
235  
240  
245  
250  
255  
260  
265  
270  
275  
280  
285  
290  
295  
300  
305  
310  
315  
320  
325  
330  
335  
340  
345  
350  
355  
360  
365  
370  
375  
380  
385  
390  
395  
400  
405  
410  
415  
420  
425  
430  
435  
440  
445  
450  
455  
460  
465  
470  
475  
480  
485  
490  
495  
500  
505  
510  
515  
520  
525  
530  
535  
540  
545  
550  
555  
560  
565  
570  
575  
580  
585  
590  
595  
600  
605  
610  
615  
620  
625  
630  
635  
640  
645  
650  
655  
660  
665  
670  
675  
680  
685  
690  
695  
700  
705  
710  
715  
720  
725  
730  
735  
740  
745  
750  
755  
760  
765  
770  
775  
780  
785  
790  
795  
800  
805  
810  
815  
820  
825  
830  
835  
840  
845  
850  
855  
860  
865  
870  
875  
880  
885  
890  
895  
900  
905  
910  
915  
920  
925  
930  
935  
940  
945  
950  
955  
960  
965  
970  
975  
980  
985  
990  
995  
1000

1162:7047-P375US

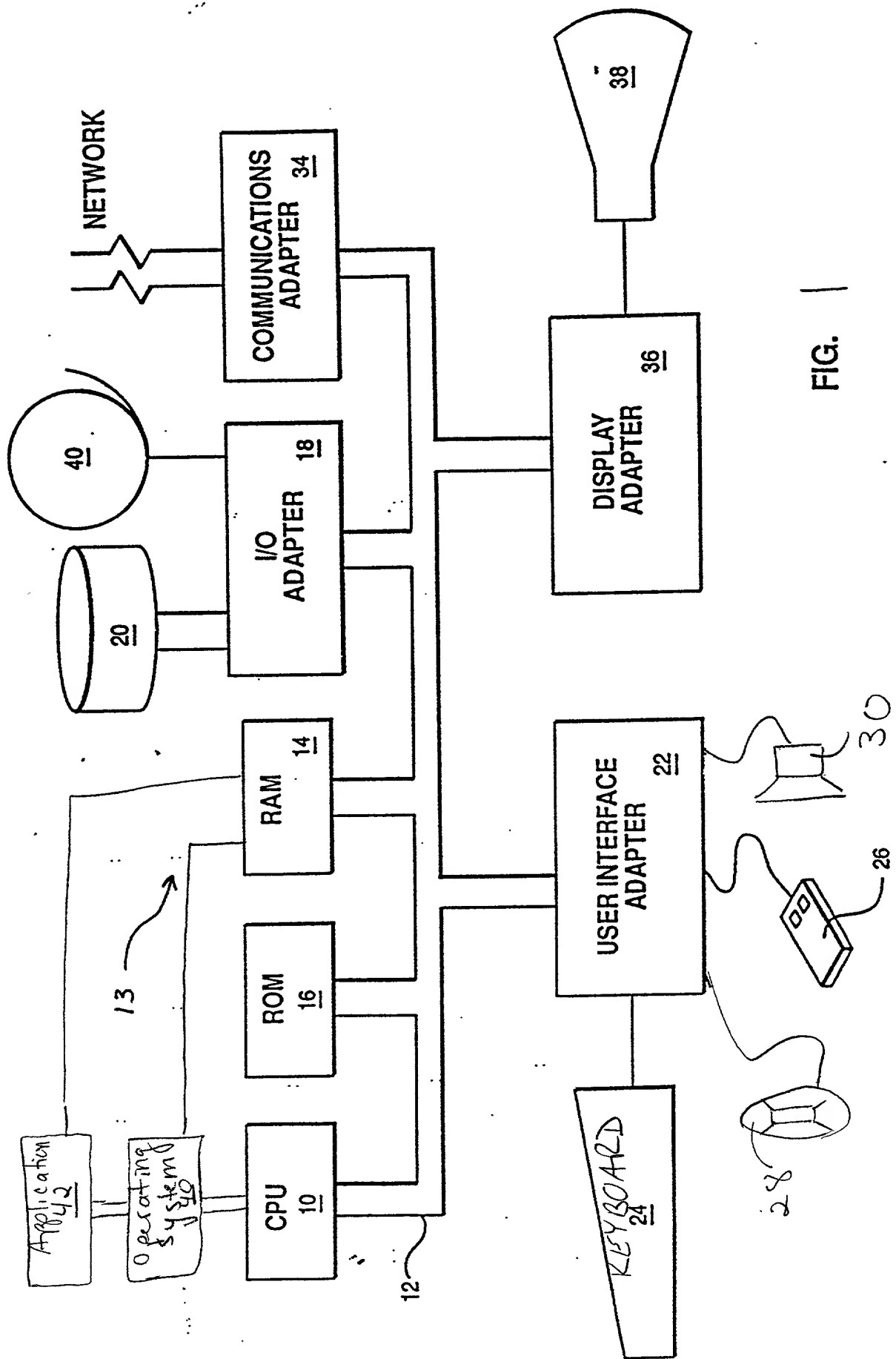
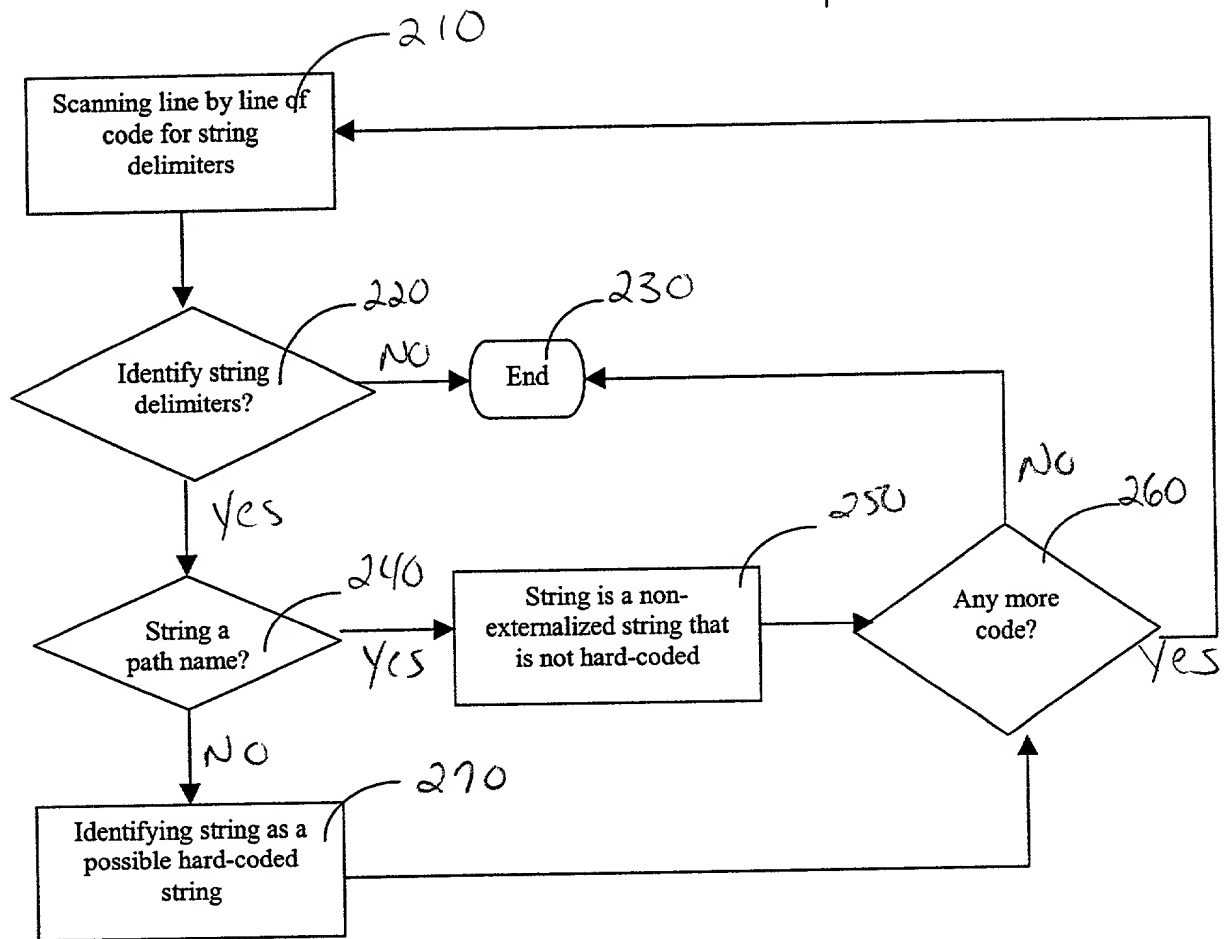


FIG. 1

Figure 2



## DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

### IDENTIFYING NON-EXTERNALIZED TEXT STRINGS THAT ARE NOT HARD-CODED

the specification of which (check one)

- ☒ is attached hereto.
- ☐ was filed on \_\_\_\_\_  
as Application Serial No. \_\_\_\_\_  
and was amended on \_\_\_\_\_

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):

Priority Claimed

_____	_____	_____	<input type="checkbox"/> Yes <input type="checkbox"/> No
(Number)	(Country)	(Day/Month/Year)	



I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

---

(Application Serial #)

---

(Filing Date)

---

(Status)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

**POWER OF ATTORNEY:** As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

John W. Henderson, Jr., Reg. No. 26,907; James H. Barksdale, Jr., Reg. No. 24,091; Thomas E. Tyson, Reg. No. 28,543; Robert M. Carwell, Reg. No. 28,499; Jeffrey S. LaBaw, Reg. No. 31,633; Douglas H. Lefevre, Reg. No. 26,193; Casimer K. Salys, Reg. No. 28,900; David A. Mims, Jr., Reg. No. 32,708; Mark E. McBurney, Reg. No. 33,114; Anthony V. S. England, Reg. No. 35,129; Volel Emile, Reg. No. 39,969; Christopher A. Hughes, Reg. No. 26,914; Edward A. Pennington, Reg. No. 32,588; John E. Hoel, Reg. No. 26,279; Joseph C. Redmond, Jr., Reg. No. 18,753; Leslie A. Van Leeuwen, Reg. No. 42,196; Marilyn S. Dawkins, Reg. No. 31,140; Kelly K. Kordzik, Reg. No. 36,571; Barry S. Newberger, Reg. No. 41,527; and Robert A. Voigt, Jr., Reg. No. 47,159.

Send correspondence to: Kelly K. Kordzik, 100 Congress Avenue, Suite 800, Austin, Texas 78701, and direct all telephone calls to Kelly K. Kordzik at (512) 370-2851.

FULL NAME OF FIRST OR SOLE INVENTOR: **DAVID BRUCE KUMHYR**

INVENTOR'S SIGNATURE:  DATE: 10/19/00

RESIDENCE: **8934 Appaloosa Run**  
**Austin, Travis County, Texas 78737**

CITIZENSHIP: **U.S.A.**

POST OFFICE ADDRESS: **(Same as Residence)**

FULL NAME OF SECOND INVENTOR: **DAE-SUK CHUNG**

INVENTOR'S SIGNATURE: \_\_\_\_\_ DATE: \_\_\_\_\_

RESIDENCE: **5700 Tapadera Trace Lane #718**  
**Austin, Travis County, Texas 78727**

CITIZENSHIP: **U.S.A.**

POST OFFICE ADDRESS: **(Same as Residence)**

::ODMA\PCDOCS\AUSTIN\_1\149932\1  
207:7047-P375US

AUS9-2000-0501-US1

**DECLARATION AND POWER OF ATTORNEY FOR  
PATENT APPLICATION**

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

**IDENTIFYING NON-EXTERNALIZED TEXT STRINGS  
THAT ARE NOT HARD-CODED**

the specification of which (check one)

- ☒ is attached hereto.
- ☐ was filed on \_\_\_\_\_  
as Application Serial No. \_\_\_\_\_  
and was amended on \_\_\_\_\_

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):

Priority Claimed

\_\_\_\_\_  
(Number)                      (Country)                      (Day/Month/Year)

☐ Yes    ☐ No

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this

AUS9-2000-0501-US1

application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

---

(Application Serial #)

---

(Filing Date)

---

(Status)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

**POWER OF ATTORNEY:** As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

John W. Henderson, Jr., Reg. No. 26,907; James H. Barksdale, Jr., Reg. No. 24,091; Thomas E. Tyson, Reg. No. 28,543; Robert M. Carwell, Reg. No. 28,499; Jeffrey S. LaBaw, Reg. No. 31,633; Douglas H. Lefevre, Reg. No. 26,193; Casimer K. Salys, Reg. No. 28,900; David A. Mims, Jr., Reg. No. 32,708; Mark E. McBurney, Reg. No. 33,114; Anthony V. S. England, Reg. No. 35,129; Volel Emile, Reg. No. 39,969; Christopher A. Hughes, Reg. No. 26,914; Edward A. Pennington, Reg. No. 32,588; John E. Hoel, Reg. No. 26,279; Joseph C. Redmond, Jr., Reg. No. 18,753; Leslie A. Van Leeuwen, Reg. No. 42,196; Marilyn S. Dawkins, Reg. No. 31,140; Kelly K. Kordzik, Reg. No. 36,571; Barry S. Newberger, Reg. No. 41,527; and Robert A. Voigt, Jr., Reg. No. 47,159.

Send correspondence to: Kelly K. Kordzik, 100 Congress Avenue, Suite 800, Austin, Texas 78701, and direct all telephone calls to Kelly K. Kordzik at (512) 370-2851.

AUS9-2000-0501-US1

FULL NAME OF FIRST OR SOLE INVENTOR: DAVID BRUCE KUMHYR

INVENTOR'S SIGNATURE: \_\_\_\_\_ DATE: \_\_\_\_\_

RESIDENCE: 8934 Appaloosa Run  
Austin, Travis County, Texas 78737

CITIZENSHIP: U.S.A.

POST OFFICE ADDRESS: (Same as Residence)

FULL NAME OF SECOND INVENTOR: DAE-SUK CHUNG

INVENTOR'S SIGNATURE: Daesuk Chung DATE: 10/23/2000RESIDENCE: ~~5700 Tapadera Trace Lane #718~~  
~~Austin, Travis County, Texas 78727~~9608 Aire Libre DR  
AUSTIN, TX 78726CITIZENSHIP: ~~U.S.A.~~ SWEDEN

POST OFFICE ADDRESS: (Same as Residence)

:\ODM\APCDOCS\AUSTIN\_1\149932\1  
207:7047-P375US